gathr

A Practical Guide to **Continuous Compliance** in DevOps

Table of Contents

Introduction

Most organizations practicing DevOps report higher productivity within their software development teams, faster product releases, and much-improved end-user experiences. However, with more frequent releases, the work of security and compliance teams has become increasingly complex and challenging. It's not easy for organizations to meet their release deadlines with all requisite security and compliance in place.

So, how to monitor and improve security and compliance across the deployment pipeline amidst tight delivery schedules? The answer lies in the "shift-left" approach, which advocates introducing quality and security tests earlier in the pipeline. The shift-left approach sounds promising; however, organizations often face challenges in its execution. It is not easy to collect and analyze data from disparate tools to meet security and compliance mandates efficiently.

Organizations need to automate all the processes to continuously track governance across the software delivery pipeline to ensure their applications are secure and compliant. We will explore a practical approach for continuous compliance that bakes security into your pipelines, relying on smart, no-code automation and DevOps best practices. We have also included a case study to demonstrate how organizations can automate their security and compliance workflows.

The Evolving **Threat Landscape**

Today, the most mature DevOps teams rely on automated CI/CD pipelines automated Continuous Integration and Continuous Deployment (CI/CD) with integrated test automation and deploy with infrastructure as code. They also have better controls and automation across their change management and incident management workflows, which allows them to detect and mitigate production issues faster. Moreover, regarding security best practices, software development teams are arguably the most well-informed.

However, it's not rare to find even the development teams responsible for a security and compliance gap, often due to a common mistake or oversight. Configuration errors, unpatched or vulnerable third-party components, inadequate encryption, and lack of access controls across source code repositories and CI/CD pipelines often lead to massive data breaches. In recent times, Log4j vulnerability and the SolarWinds supply chain attacks have highlighted the importance of improving software supply chain security. Malwares have been known to take the guise of security updates, hiding behind trusted DevOps tools, compromising enterprise security, and causing costly breaches and penalties. The recent Circle CI breach highlights that even your (CI/CD) platform can be targeted by threat actors.

An average application development project has **49 vulnerabilities and 80 direct dependencies** on open-source software.

— Snyk

66% of respondents in a survey said that application security **tools protect less than 75% of their codebase;** 48% acknowledged that **they push vulnerable code into production** regularly.

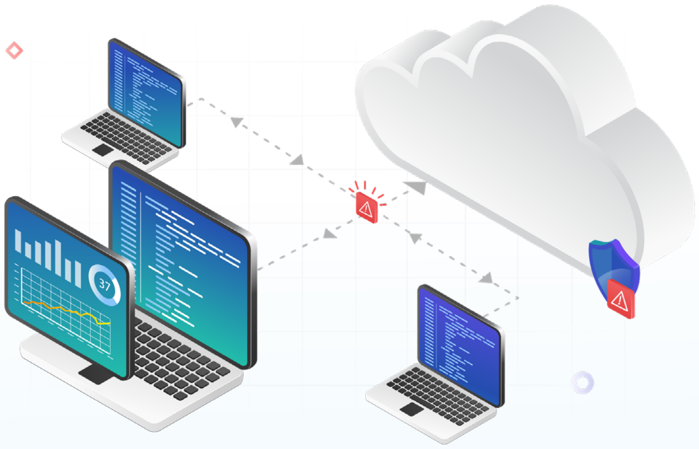
— Modern Application Development Security Research, ESG

There's no lack of security tools to counter such threats. However, Chief Information Officers (CIOs) and DevSecOps teams realize that improving the governance across DevOps and better management of open source and commercial components can help them fix the most glaring security and compliance issues. As a result, organizations are looking for ways to automate policy actions such as Software Analysis Security Testing (SAST) as part of the CI/CD pipeline, assigning minimum privileges to access source code repositories, scanning code for vulnerabilities, encrypting connections, and more.

Policy-as-Code has emerged as one of the ways to simplify such automation. Policy-as-code, as the name suggests, involves codifying and enforcing policies that previously required manual workflows. Often seen as an extension of infrastructure as code, policy-as-code goes beyond infrastructure provisioning and can be used for authorization and access controls for Application Programming Interfaces (APIs), Kubernetes resources (pods, nodes, clusters), compliance and auditing (Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), General Data Protection Regulation (GDPR), etc.), and more.

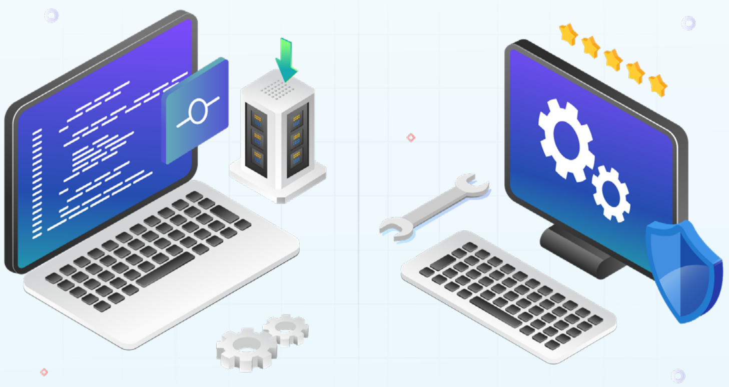
Security & Compliance **Challenges in DevOps**

Keeping Up with Deliveries



DevOps has largely evolved with Lean principles – maximize efficiency, eliminate waste and delays, and cut unnecessary costs. There's an emphasis on failing fast and failing early. With rapid prototyping (Minimum Viable Products or MVPs), organizations want to get early feedback from users in the production environment and make iterations to improve. This puts a question mark on their investments in security; why and how much time and effort should they apply to secure code that's likely to be discarded or rewritten in the next few days? Nonetheless, industry leaders, these days make multiple daily deployments, sometimes deploying every second. With so many changes happening across teams and projects, it's difficult for security teams to keep up. They rarely have the time to gather data, analyze it, and assess risks and are forced to release code to production without a security or hardening sprint.

Separation of Duties in DevOps

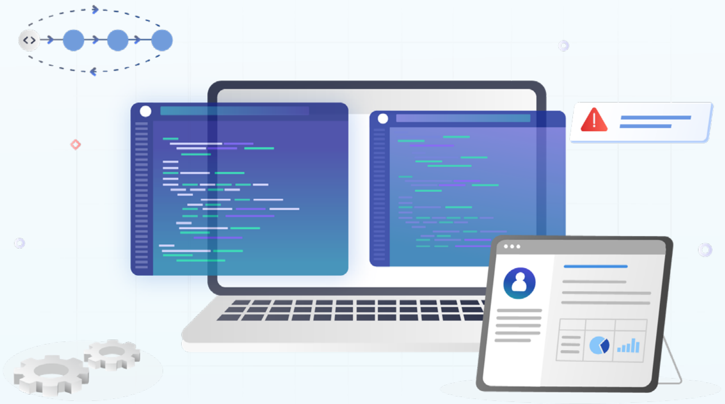


Security and governance frameworks (ITIL, ISO 27001, NIST 800-53, etc.) and compliances (PCI DSS, Sarbanes-Oxley Act (SOX), Gramm-Leach-Bliley Act (GLBA), etc.) refer to Separation of Duties (SoD) as a critical control. By delineating the duties of ops and development, organizations can easily manage and control risks and insider threats with clear audit trails. DevOps can appear to conflict with such controls as it advocates sharing responsibilities between the teams. It is not easy for security and compliance teams to monitor developers' activity in a production environment, track every code or configuration change, identify changes made outside of the continuous delivery pipeline, and sift through endless streams of logs to detect accidental leakage of confidential data.

Cloud & Container Complexities

While CloudOps and DevOps are increasingly becoming inseparable, many organizations have a hard time keeping up with the security practices for the cloud. In the shared responsibility model for the cloud, organizations are often caught off guard, failing to account for where the cloud providers' responsibilities end and theirs begin. Insecure interfaces and APIs, weak identity, credentials, and access management, account hijacking, malicious insiders, or privilege abuse often cause cloud data breaches. While containerization has made the lives of developers many times simpler, solving the classic "it runs on my machine" problem, it has also created a fair share of new DevOps challenges. Teams now need to keep track of Kernel vulnerabilities, container breakouts, poisoned images, and more. Organizations are seeking ways to enforce the usage of trusted registries and image scanning to improve container security.

Change Management



The conventional Change Advisory Board (CAB) seems to have lost relevance in the DevOps world. How can they function when developers push code directly to production multiple times a day? Information Technology Infrastructure Library (ITIL) change management and similar practices designed to manage big changes in the past aren't suited for modern development schedules. Moreover, not all organizations have made the transition smoothly and have failed to update their compliance and risk management practices to live up to DevOps expectations.

The answer lies in the emerging continuous compliance or automated governance practices that tie governance and change monitoring with CI/CD.

What is Continuous Compliance?

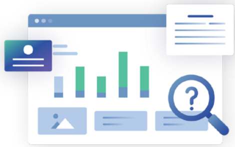
Continuous Compliance (or Automated Governance) is a set of practices that aims to automate information security, audit compliance, and change management to keep up with modern delivery schedules. These practices can replace traditional manual governance models that aren't able to keep up with the high volume of daily changes. By making governance a part of CI/CD and eliminating manual hand-offs from commit to production, organizations practicing continuous compliance can avoid all workarounds or shortcuts that developers take to meet their deadlines while inadvertently exposing applications to risk and vulnerabilities.

While people generally associate compliance with increased documentation and regulatory nuisance, it's essential to understand that continuous compliance in DevOps can help teams achieve higher security and agility. For example, making vulnerability scans a part of the CI/CD process and running them automatically at build time can help teams detect vulnerabilities earlier in the cycle.

In the context of information security, continuous compliance can involve automated monitoring of system logs, software configurations, licensing compliances, cloud platforms and services, user and entity behavior analysis, adherence to security best practices, benchmarks, frameworks, and more. Continuous compliance can also help enforce Identity and Access Management (IAM) policies, which are vital for compliance. Such policies help control access to data by defining access rights for teams and roles instead of individuals. Auditing and change management can also benefit from continuous compliance. It makes it easier to trace a vulnerability to all related changes, commits, and lines of code. Organizations can automate and enforce policies that ensure peer reviews, test coverage, and other controls are in place and that the violations are easily detected.

Continuous Compliance **Implementation**

Gathr frequently engages with DevOps teams, helping them solve their process challenges. We will describe one of our continuous compliance implementations, which was part of a proof of concept Gathr conducted with a leading US-based fintech company. The company was facing multiple challenges in its software development projects due to ineffective compliance monitoring:



Lack of centralized visibility to assess compliance across enterprise projects

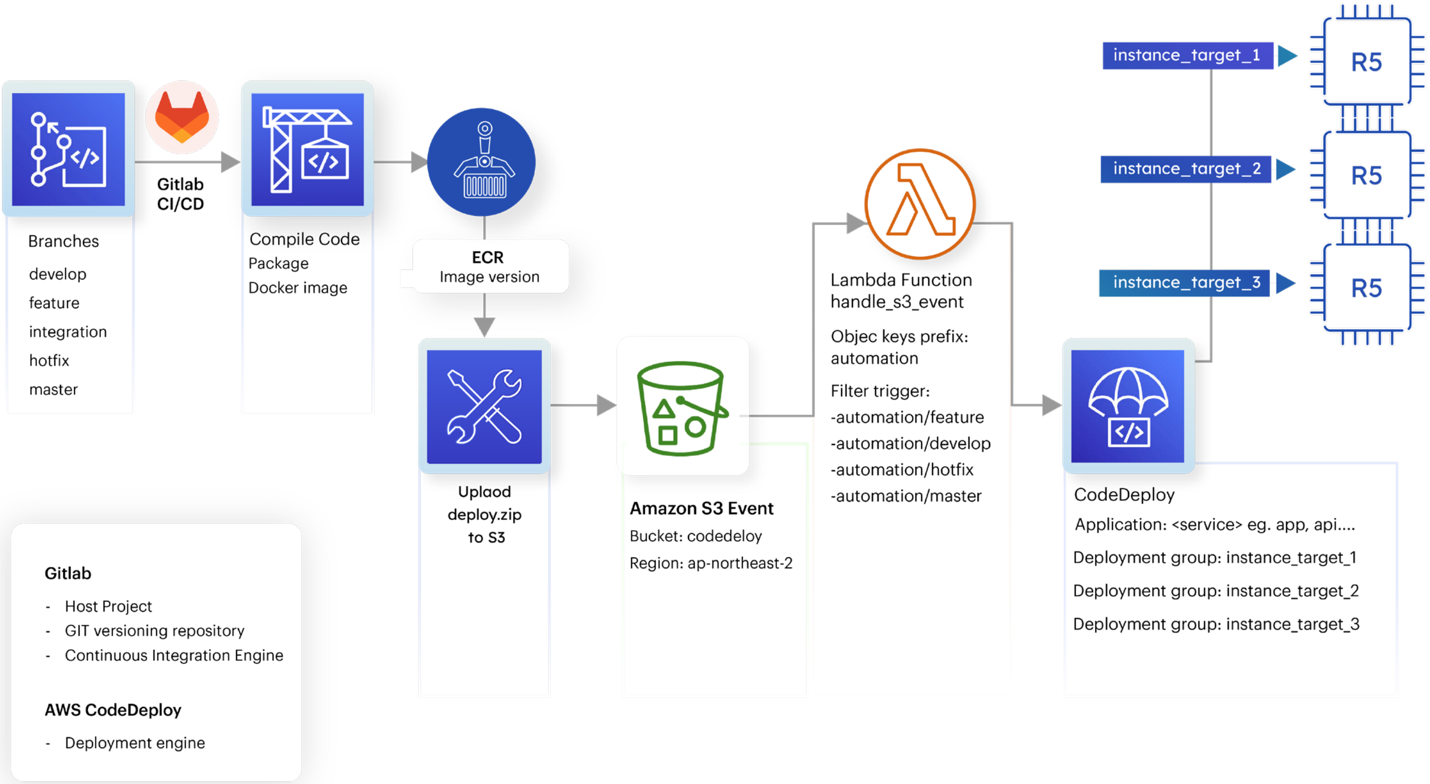
Manual tracking of every commit, pull request (PR), and peer approval was untenable

It wasn't easy to track if developers used the pre-defined tools and procedures for version control, source code management, peer reviews, etc.

The **Solution**

Gathr helped the company with a flexible approach to monitoring changes across a delivery pipeline, automating compliance, introducing release gates, and ensuring secure and reliable releases. Gathr's smart connectors helped in the collection and visualization of data from tools like GitLab and Amazon Code Deploy. The integration with AWS CodeDeploy was required to get the ID of commits, which helped in tracing them back to the GitLab commit. The following data points were used for compliance tracking:

- Commits and modified files
- Merge Requests (to merge commits into master branch)
- Merge Requests Comments (to identify Code Review)
- Builds and Deployments: Pipelines that execute jobs



Gitlab

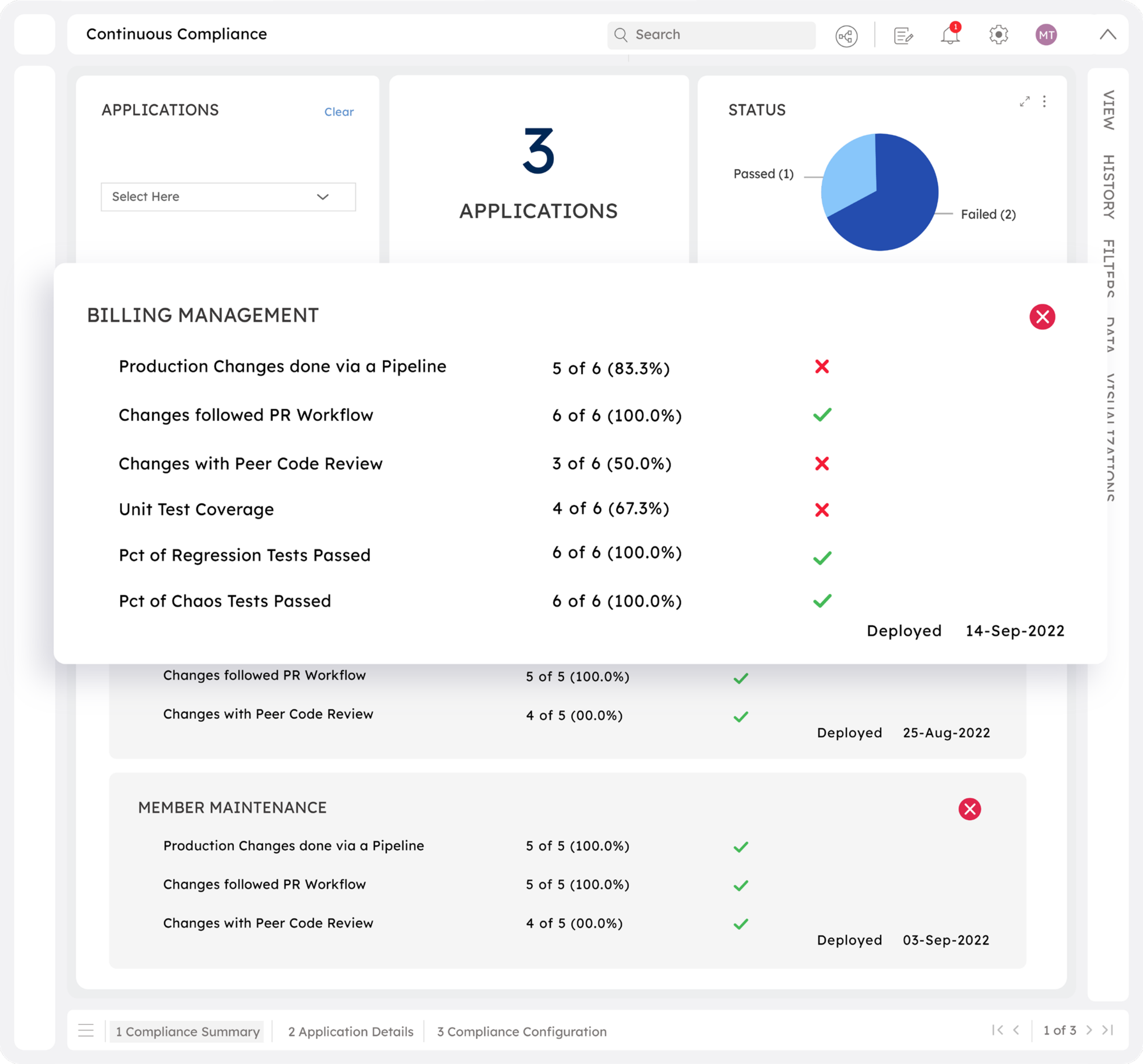
- Host Project
- GIT versioning repository
- Continuous Integration Engine

AWS CodeDeploy

- Deployment engine

Quick Summary

After integrating the tools and data, Gathr created a summary dashboard for executives to offer a quick overview of compliance across projects.



BILLING MANAGEMENT

Production Changes done via a Pipeline	5 of 6 (83.3%)	✗
Changes followed PR Workflow	6 of 6 (100.0%)	✓
Changes with Peer Code Review	3 of 6 (50.0%)	✗
Unit Test Coverage	4 of 6 (67.3%)	✗
Pct of Regression Tests Passed	6 of 6 (100.0%)	✓
Pct of Chaos Tests Passed	6 of 6 (100.0%)	✓

Deployed 14-Sep-2022

Changes followed PR Workflow	5 of 5 (100.0%)	✓
Changes with Peer Code Review	4 of 5 (00.0%)	✓

Deployed 25-Aug-2022

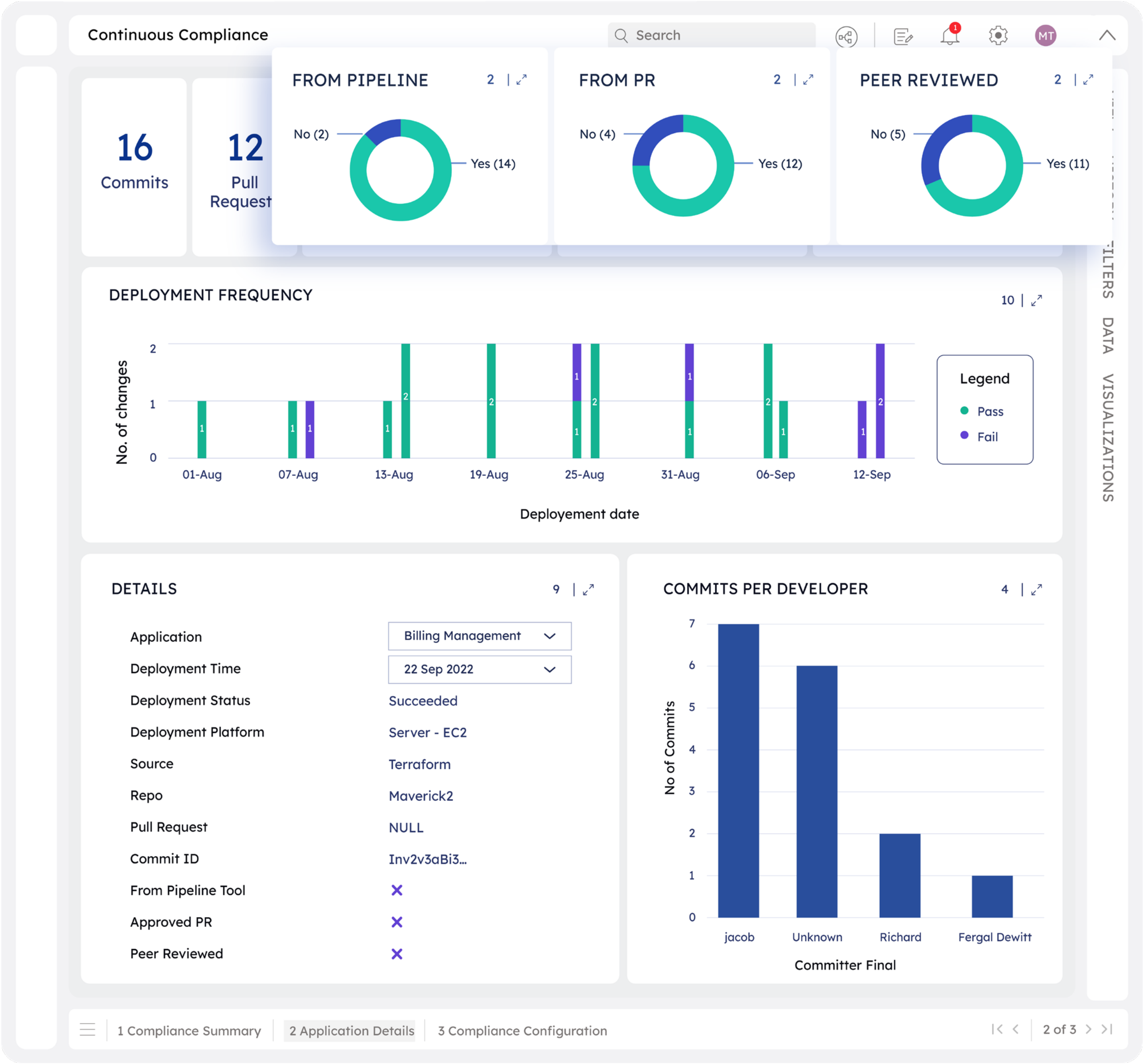
MEMBER MAINTENANCE

Production Changes done via a Pipeline	5 of 5 (100.0%)	✓
Changes followed PR Workflow	5 of 5 (100.0%)	✓
Changes with Peer Code Review	4 of 5 (00.0%)	✓

Deployed 03-Sep-2022

Detailed Breakdown

It also enabled visual tracking of the percentage of changes following the pipeline tools, PR approvals, and peer reviews. The dashboard offered contextual details for root cause analysis and release management purposes.

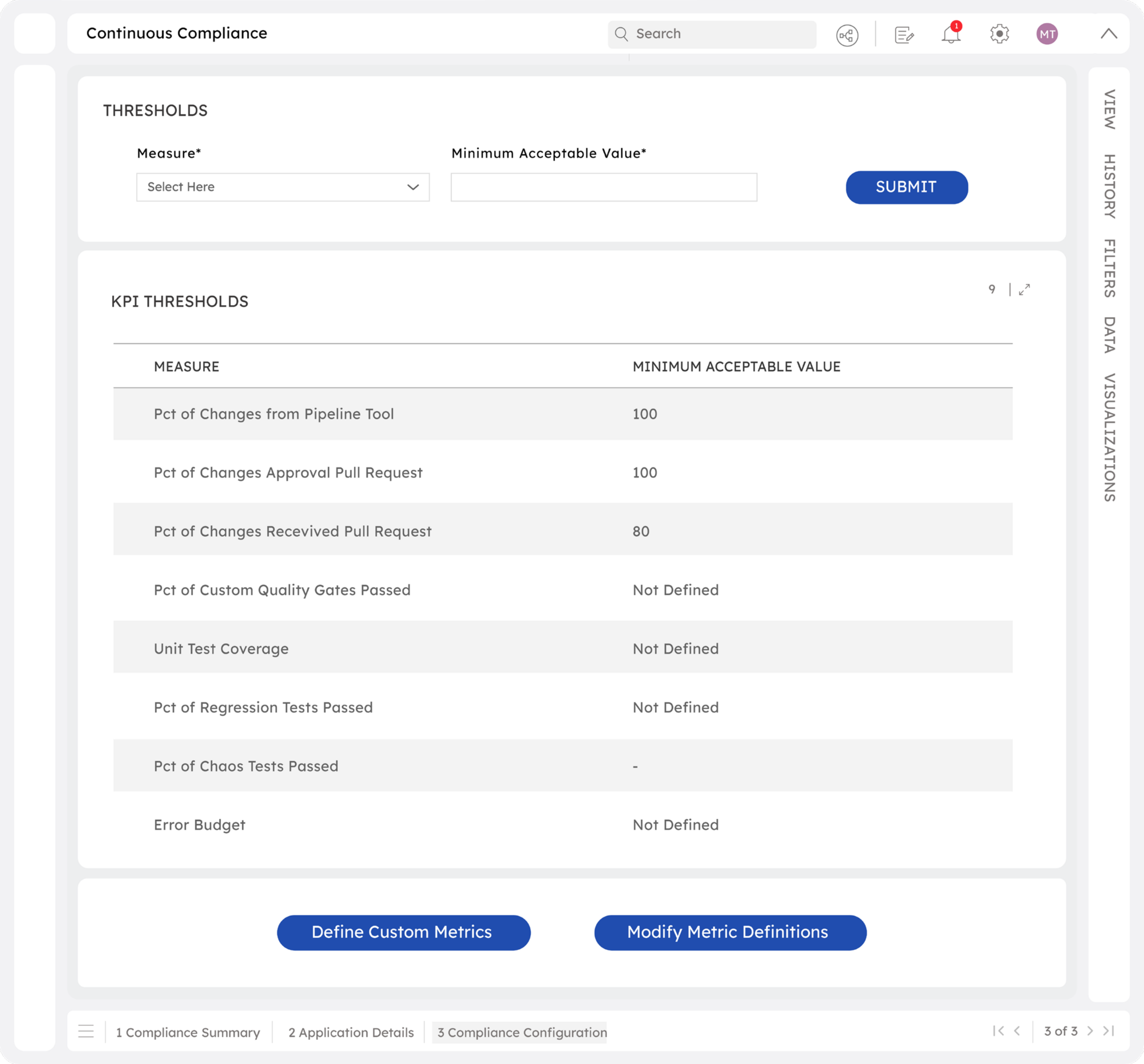


DETAILS

Application	Billing Management
Deployment Time	22 Sep 2022
Deployment Status	Succeeded
Deployment Platform	Server - EC2
Source	Terraform
Repo	Maverick2
Pull Request	NULL
Commit ID	Inv2v3aBi3...
From Pipeline Tool	✗
Approved PR	✗
Peer Reviewed	✗

Configuration & Customization

Further, the solution made it possible to configure or adjust application pass/fail criteria with different metric thresholds to meet varying compliance needs across different projects and teams. The teams could use the configuration flexibility for what-if analysis to test different scenarios.



THRESHOLDS

Measure*: Select Here

Minimum Acceptable Value*:

KPI THRESHOLDS

MEASURE	MINIMUM ACCEPTABLE VALUE
Pct of Changes from Pipeline Tool	100
Pct of Changes Approval Pull Request	100
Pct of Changes Recevived Pull Request	80
Pct of Custom Quality Gates Passed	Not Defined
Unit Test Coverage	Not Defined
Pct of Regression Tests Passed	Not Defined
Pct of Chaos Tests Passed	-
Error Budget	Not Defined

The **Impact**

After a proof of concept, the company quickly onboarded Gathr's continuous compliance solution. It found the solution highly effective in assessing the completeness of changes and pipeline compliance. Instead of wasting hours, the teams could trace every change to its author, reviewer, and approver in minutes to identify compliance gaps and training needs.

> We have seen an increase in pipeline compliance by 3X within a month of implementing Gathr's solution. With better CI/CD compliance, we expect continuous improvements in quality in the future.
>
> — Senior Director, DevOps

How to Extend **Continuous Compliance** Across DevOps

It is possible to extend the continuous compliance implementation described in the previous section to ensure the CI/CD process generates transparent audit trails of critical actions, approvals, and controls. Gathr enables quick tool integration and workflow automation, which can allow organizations automate compliance and governance while ensuring the following:

- Product owners authorize all changes.
- Peers review the code for quality and validity.
- All changes are tested and meet exit criteria.

- A deployment tool validates that code being migrated to production has met quality standards and requisite quality controls. Teams can include application security checks as part of the automated process. Software Composition Analysis (SCA), Dynamic Application Security Testing (DAST), Static Application Security Testing (SAST), and Interactive Application Security Testing (IAST) are some of the common tests teams can consider for their pipeline.
- All tools in the pipeline have Information Technology Change Controls (ITGCs) (change control, role-based access controls, segregation of duties).
- The control environment is continuously monitored for effectiveness. Changes without controls evidence trigger remediation.

Teams can also leverage Gathr to develop custom compliance solutions bringing planning and operations stages into its coverage. To this end, organizations must first define controls or compliance policies for different stages.

Gathr can solve your lingering security and compliance challenges with a unique no-code platform that expedites tool integration, workflow automation, insights delivery, and data analytics. Learn more about our DevOps solutions or sign up for a demo now.